

### Amendments to the Drawings

Please amend drawing Figures 8, 12, 15, 16, 18, 20 and 21 as requested by the Examiner by entering the seven Replacement Sheets submitted herewith.

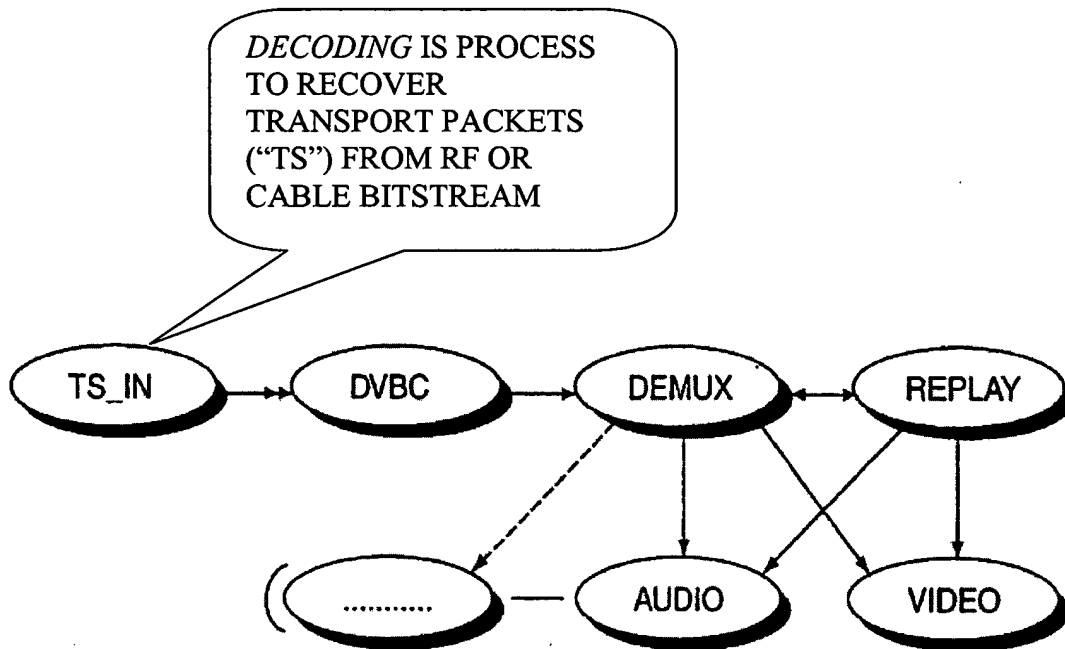
### Remarks

This amendment is submitted in response to the non-final Office Action mailed 02/08/2005. Drawing corrections are submitted, by way of seven replacement sheets, to address the objections made by the Examiner with regard to Figures 8, 12, 15, 16, 18, 20 and 21. These replacement sheets are believed to obviate the objections.

Claims 1-34 were variously rejected as anticipated by or obvious in view of the prior art. Applicant respectfully traverses the rejections and requests reconsideration for the reasons explained below. In this reply, claims 11 and 25-34 are amended. New claims 35-38 are submitted to more completely claim the invention.

### Discussion

Claims 1-3, 11-13, 17-19, and 25-27 were rejected as anticipated by Nooralahiyan et al. (US 6,775,463 B1). Applicant respectfully traverses these grounds for rejection and requests reconsideration in view of the foregoing amendments to the claims and the comments below. Nooralahiyan discloses a video receiver and display method reminiscent of the currently popular "TiVo" system. That is, it enables selected replay of broadcast audio and video content, at normal or slow motion speeds, on demand. This is accomplished by implementing physical memory buffers to store data packets. The data stored in the buffers has already been decoded into usable video content. To illustrate, below is Nooralahiyan Fig. 1 with a bubble callout comment added:

**FIG 1**

Nooralahiyan explains the video replay process at the top of column 4:

"The interaction between the DEMUX routing process (producer) and the VIDEO process (consumer) utilises a short buffer of few data packets. This is to say that as soon as these few packets of video data are available, they are routed to the video process. This, in turn, interacts with the video driver. Thus, the producer pointer is constantly moving round the circular buffer and is followed closely by the consumer pointer as illustrated in FIG. 2 to define a first portion of the memory in accordance with the invention. The dotted line 2 consists of previously decoded and displayed video packets and is referred to as the second or remainder portion of the memory in accordance with the invention also referred to as the history loop. When the receiver or set top box (STB) is instructed to operate the video replay mode such as by the reception of a signal from a remote control, the REPLAY process first interrupts the DEMUX process and halts the subsequent routing of the video data packets. The REPLAY

process then takes over the task of the DEMUX/Routing process and channels the video packets in residence in the history loop (in the memory) to the VIDEO process.”

“The implementation of video replay is achieved by accessing the video packets in the history loop and the packets are subsequently interfaced to the video driver.”

The present invention *has nothing to do with* “video replay” as that term is used in Nooralahiyan. Rather, the present invention is about sharing decoders (e.g., Viterbi decoders) to support substantially simultaneous decoding of multiple broadcast channels, using a number of decoders that is fewer than the number of broadcast channels. “In one embodiment, eight Viterbi decoders 700 are shared across thirty-two transponders with each Viterbi decoder servicing four transponders.” See Specification, page 42, line 12. Nooralahiyan says nothing about sharing a decoder to decode multiple content channels or sources. It is about buffering a single, already-decoded video stream for selected replay.

By contrast, claim 1 of the present invention, for example, is directed to a “method for sharing a decoder among a plurality of data streams.” Sharing decoders requires a “context switch” whenever a decoder switches from decoding one channel or stream to another:

“In one embodiment, when a particular decoder changes the transponder or cable carrier it is processing, the decoder is restored to the same state it was in when it was processing the same transponder/carrier the last time around. As used herein, a “context switch” occurs when the decoder switches from processing one transponder/carrier to another.”

“Restoring the state of the decoder on a context switch may be accomplished in a variety of ways. For example, in one embodiment, the decoder state is saved on every context switch. The correct state is then selected when processing a particular transponder. [*This is called the non-replay scheme.*] Alternatively, or in addition, the state itself may not be saved. Rather, when switching from one transponder (or cable carrier) to another, N symbols output by the transponder the last time around are

replayed, thereby restoring the state of the decoder. [*This is called the replay scheme for context switching.*]"

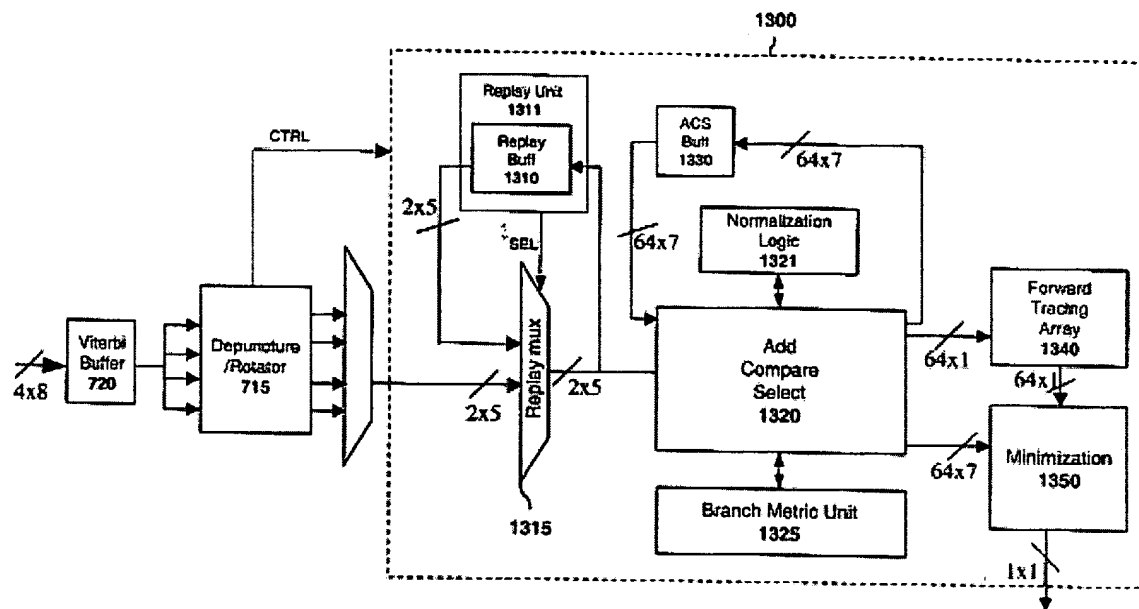
Thus, the "replay scheme" of the present invention – referring to digital decoder context switching – is actually unrelated to "video replay" described in Nooralahiyan. The replay scheme for context switching is described in one embodiment at pages 44-46. For example:

"One embodiment of a Viterbi decoder 1300 which employs a replay scheme is illustrated in FIG. 13a. As used herein, a "context switch" refers to a switch from processing data from one transponder (or other signal carrier--e.g., such as a cable carrier) to another. Using the replay scheme, on a context switch, the forward-tracing array 1340 may be restored to the state it was in when it left off processing the same transponder the last time around. In one embodiment, this is accomplished by replaying the last N symbols of the transponder through the decoder and discarding the output. After the last N symbols are replayed, the decoder is ready to accept new symbols for the transponder/carrier." Specification, paragraph bridging pages 44-45 (emphasis added).

More specifically, the Examiner states, with regard to claim 1, that Nooralahiyan teaches ... "replaying data samples stored in said replay buffer to restore decoder to state it was in when it last decoded samples from data stream prior to processing new data samples (see column 3, lines 21-27 where this replay functionality is interpreted as equivalent)." Office Action at page 3, section 2. For the reasons explained above, Nooralahiyan's video replay functionality is in no way equivalent, or even remotely similar, to the replay scheme for decoder context switching of the present invention. While terms like "replay mux" and "replay buffer" appear in the present specification, these are part of the decoder itself, as illustrated below, and the "data" they store and manipulate are transponder symbols, not actual (decoded) video data like that stored in the circular buffer in Nooralahiyan. Transponder symbols may be, for example, QPSK modulated I/Q pairs. See paragraph [0140]. (See also Haas, et al. discussed below.) In

short, Nooralahiyan describes buffering already decoded video data, i.e., transport packets, whereas the present invention is about the decoding process itself.

To illustrate, Fig. 13a shows a Viterbi decoder 1300 that employs a replay scheme for context switching.



**Fig. 13a**

As shown above, transponder symbols are “replayed” through the Viterbi process to restore the decoder state. In one preferred embodiment, N symbols are replayed, where N is equal to or greater than the depth of the Viterbi trellis. Importantly, while video content is “replayed” using Nooralahiyan’s video replay system for viewing by a user, the decoder output generated during “replay scheme” context switching in accordance with the present invention is tossed out: “After the last N symbols are replayed, the decoder is ready to accept new symbols for the transponder/carrier. During the replay period, the output from the forward-tracing array 1340 may be ignored.” Specification, paragraph bridging pages 44-45.

Similarly, with regard to claim 2, Nooralahiyan’s circular buffer cannot be “interpreted as equivalent functionality” because the context is non-analogous, and specifically because Nooralahiyan’s buffer stores decoded video data packets, not encoded data symbols. Nooralahiyan explains clearly, “The dotted line 2 consists of previously decoded and displayed video packets and is referred to as the second or

remainder portion of the memory in accordance with the invention also referred to as the history loop.” Column 4, line 9 (emphasis added).

Nooralahiyan explains, with reference to their Fig. 1, “The TS-IN process compares the Packet ID (PID) of the incoming Transport Packet with the PID Action Table held in the memory of the receiver to distinguish between packets that should be routed to the memory and the packets of data which can be rejected.” Col. 3, lines 47-49. The term “*Transport Packet*” is well-known in DVB to comprise bits of actual video data, for example see MPEG-2 Transport Packets. At the transmitter end, i.e. before transmission, the Transport Packets of data are assembled into frames; synchronization bytes inverted; contents scrambled, R-S coding bits added, interleaved convolutional coding; and the resulting bit stream modulated, for example using QPSK modulation. At the receiver, as is well-known, all of that is “un-done” to recover the Transport Packets of data. (See Haas et al. discussed below.) The present invention relates to that decoding process. By contrast, Nooralahiyan discloses buffering and displaying the recovered transport packets of data, e.g., MPEG-2 audio and video data bytes.

With regard to claim 11, for the reasons explained above, applicant respectfully disputes the Examiner’s position set forth at page 4 of the Office Action. Claim 11 is amended in the preamble to clarify that it is directed to a “replay method of context switching a decoder.” There is nothing in Nooralahiyan even resembling, for example, the limitation of, “restoring said decoder to said first state by re-decoding said first set of data from said buffer.” For at least these reasons, the rejection of claim 11 and those claims depending from it should be reconsidered and withdrawn.

With regard to system claims 17-19, the Examiner would apply by analogy the asserted grounds for rejection of method claims 1-3. Applicant respectfully traverses the rejection. The claim reads:

“17. (Previously presented) A system comprising:

a decoder for decoding data from a plurality of data streams;

data replay means for restoring said decoder to a state it was in when it previously decoded data from each respective data stream, before decoding new data from each respective data stream.”

This claim cannot be anticipated by Nooralahiyan, as Nooralahiyan discloses no decoder at all, and Nooralahiyan discloses no means at all for “restoring said decoder to a state it was in when it previously decoded data from each respective data stream.” The dependent claims should be allowed as well.

Regarding claim 25 and those claims that depend from it, the claim is amended to delete reference to stored code, and thus now refers directly to an integrated circuit. Moreover, the claimed integrated circuit of claim 25 comprises, in part:


“a decoder for decoding data ~~samples~~ symbols from a first data stream; [and]  
a replay buffer for storing N data ~~samples~~ symbols processed from said first data stream in a replay buffer before decoding data from other data streams....”

The term “symbols” is used in the specification, consistent with ordinary usage in the field, to refer to modulated bits, one example being QPSK modulation, in which four bits are modulated into one symbol. (Haas et al. discloses a QPSK chipset.) Nooralahiyan discloses nothing about demodulating or decoding symbols. For at least these reasons, the rejection of claim 25 and those depending from it should be withdraw. Dependent claims 26-34 are amended for consistency in the preambles with the base claim.

The Examiner also relies on Haas et al. “Advanced Two IC Chipset for DVB on Satellite Reception” (IEEE Trans. On Consumer Electronics, Vol. 42, No. 3, August 1996). Haas et al. disclose a bipolar I/Q demodulator chip and an MOS QPSK “processor device” with ADC, clock and carrier recovery, Viterbi decoder and RS forward error correction. It outputs a single stream of recovered MPEG data (see Fig. 3). No disclosure is found in Haas regarding context switching of the Viterbi among multiple input steams. The Examiner correctly observes that Nooralahiyan is silent on specifics of his DVBC block. Haas describes a DVB system, but neither reference nor the combination discloses or suggests the features of the present claims.

In view of the foregoing amendments and remarks, the present application is now believed in condition for allowance. The Examiner is encouraged to telephone the undersigned if any issues remain.

Respectfully submitted,

By   
Kory D. Christensen  
Registration No. 43,548

STOEL RIVES LLP  
One Utah Center Suite 1100  
201 S Main Street  
Salt Lake City, UT 84111-4904  
Telephone: (801) 328-3131  
Facsimile: (801) 578-6999